1980

Apple I/O card documentation

P. J. McKerrow
University of Wollongong, phillip@uow.edu.au

APPLE I/O CARD DOCUMENTATION

By

Phillip McKerrow

Preprint No. 80-8

APPLE I/O CARD DOCUMENTATION

1. Specifications of Card

8 digital outputs, TTL level, sink 40mA, source 10mA
8 digital inputs, TTl level, maximum 20μA
1 Analog output 0-10V, 5mA maximum output current
1 Analog input 0-10V, 3KΩ minimum input impedance

Variations

1. digital outputs can be used as digital inputs

2. Analog output variable between ±10V by changing reference voltage and reference resistor.

3. Analog input can be changed to -5, -0, -+5V by removing link Z.

2. Bibliography

1. Terrapin Turtle Instruction and Assembly Manual.

2. Controlling a Turtle with an Apple (1980), Phillip McKerrow.

3. Motorolla M6800 Microcomputer System Design Data.

4. Analog Devices Data Acquisition Products Catalogue.

3. Installation

The card can be inserted into any of the I/O slots on the Apple mother board. The attached programs are written for slot 2. The components go toward the right side and I/O connectors toward the front. Plug I/O cable into card before installing into the Apple.

N.B. Power should be off when card is inserted or removed.

4. Addressing

a. Base Address (A)

Slot No.	Hex Address	Decimal Address
Ø	CØ8Ø	-16256
1	CØ9Ø	-16240
2	CØAØ	-16224
3	CØBØ	-162Ø8
4	CØCØ	-16192
5	CØOØ	-16176
6	CØEØ	-1616Ø
7	CØFØ	-16144

b. Input/Output Address

Address	Function
A + Ø	Data Register A
A + 1	Control Register A
A + 2	Data Register B
A + 3	Control Register B
A + 8	Analog Input

5. PIA Initialization and Digital I/O

a. A side Digital Out

```
10   Poke  A + 1,  Ø        Select Data Direction Register
20   Poke A,  225           Select lines as outputs
30   Poke A+1,  52          Select Data Reg and Enable Buffer
40   Poke A,  Data          Output Data
```

b. A side Digital In

```
10  Poke  A + 1,  Ø          Select Data Direction Register
20  Poke  A,  Ø              Select lines as Inputs
30  Poke  A + 1,  6Ø         Select Data Reg and Disable Buffer
40  DATA = PEEK(A)           Read Data
```

c. B side Digital In

```
10  Poke  A + 3,  Ø          Select Data Direction Register
20  Poke  A + 2,  Ø          Select lines as inputs
30  Poke  A + 3,  6Ø         Select Data Reg & Enable Buffer
40  DATA = PEEK(A + 2)       Read Data
```

6. Analog I/O

a. Initialize PIA, A side as required for Digital Signals, B side as Digital in.

b. Analog In

```
10  Poke  A + 3,  52         Start conversion
20  X = PEEK(A + 3)          Read status
30  IF X < 128 then 20       Is conversion complete
40  Input = PEEK(A + 2)      Read Data
50  Poke  A + 3,  6Ø         Finish conversion
```

N.B. 1. Between the start and finish of conversion (CB2 Low) the Analog Input has control of the B Side Data Lines and the B Side Digital Input Buffer is disabled.

2. CB2 going low starts the conversion.
CB2 high tristates the ADC outputs

3. CB1 going low indicate conversion complete.

4. CB2 is controlled by Control Register B bits 5, 4, 3
11Ø = low 111 = high

5. The status of CB1 can be read from Control Register B bit 7. It will be set (high) when CB1 input signal goes low if Control Register B bits 1 and Ø are set to zero. It can also be set to interrupt the processor when the conversion is complete.

b. Analog Out

```
10  Poke  A + 8,  Data          Write data to address
```

7. Turtle Cable - from I/O Card to Molex Connector

Connector A (f)			Molex Plug	
Pin	Data	Signal	Pin	Colour
1	RMF	PAØ	11	Black
2	RMB	PA1	10	Brown
3	LMF	PA2	4	Red
4	LMB	PA3	7	Orange
5	Light	PA4	1	Yellow
6	Pen	PA5	6	Green
7	Horn	PA6	14	Blue
7	Tone	PA7	2	Violet

Connector B (n)			Molex Plug	
Pin	Data	Signal	Pin	Colour
1	LT	PBØ	3	Black
2	RT	PB1	12	Brown
3	FT	PB2	13	Red
4	BT	PB3	9	Orange
	+12V (from 12V supply		5	grey/black
9	0V (from 12V supply & I/O card)		8	white and grey

grey wire from I/O card not connected

```
LIST
   10 PRINT "APPLE I/O TEST"
   20 PRINT "P.J.MCKERROW.....5.3.80"
   40 C=0
   50 W=-16224
   60 R=-16222
   70 REM  C=ERROR COUNT,I=TEST COUNT
   80 REM  W=WRITE ADDRESS,R=READ ADDRESS
   90 REM  X=DATA OUT,Y=DATA IN,T=TEMP
   91 PRINT
   92 PRINT "TEST  ERROR       DATA"
   96 PRINT "                 OUT   IN"
  100 FOR I=1 TO 100
  105 GOSUB 300
  110 FOR X=1 TO 255
  120 POKE W,X
  130 Y= PEEK (R)
  140 IF Y=X THEN 150
  142 C=C+1
  144 PRINT I,C,X,Y
  150 NEXT X
  160 PRINT I,C
  170 NEXT I
  180 GOTO 400
  300 REM  INITIALISE PIA
  310 POKE W+1,0
  320 POKE W,255
  330 POKE W+1,52
  340 POKE R+1,0
  350 POKE R,0
  360 POKE R+1,60
  370 RETURN
  400 END
```

```
LIST
   10 PRINT "APPLE ANALOG I/O TEST"
   20 PRINT "  P.J.MCKERROW....7.3.80"
   80 REM  INITIALISE PIA
   90 PRINT "FOR IO SLOT 2"
   91 REM  A=ADDRESS,V=MAX VOLTS OUT
   92 V=10
   95 A=-16224
  100 POKE A+1,0
  110 POKE A,255
  120 POKE A+1,52
  130 POKE A+3,0
  140 POKE A+2,0
  150 POKE A+3,60
  160 PRINT "   ANALOG  DATA"
  170 PRINT "OUT      IN   VOLTS OUT *10"
  180 O=1
  240 GOSUB 2000
  250 GOSUB 1000
  260 O=2*O
  270 IF O=256 THEN O=255
  280 IF O<256 THEN 240
  300 O1=1
  310 O=255-O1
  320 GOSUB 2000
  330 GOSUB 1000
  340 O1=2*O1
  350 IF O1<255 THEN 310
  400 END
 1000 REM  WAIT LOOP
 1010 T=0
 1020 T=T+1
 1030 IF T<500 THEN 1020
 1040 RETURN
 2000 REM  I/O ROUTINE
 2010 POKE A+8,O
 2020 POKE A+3,52
 2030 X= PEEK (A+3)
 2040 IF X<128 THEN 2030
 2050 I= PEEK (A+2)
 2060 POKE A+3,60
 2070 VO=10*O*V/255
 2080 PRINT O,I,VO
 2090 RETURN
```

```
LIST
   10 PRINT "         TURTLE TEST"
   20 PRINT
   30 PRINT "SUPPORT THE TURTLE SO THAT THE WHEELS ARE FREE TO TURN"
   90 GOSUB 1000
  100 W=-16224
  110 R=-16222
  190 REM  INITIALISE PIA
  200 POKE W+1,0
  210 POKE W,255
  220 POKE W+1,52
  230 POKE R+1,0
  240 POKE R,0
  250 POKE R+1,60
  290 PRINT
  300 PRINT "      OUTPUT TEST"
  310 PRINT
  320 PRINT "TEST ORDER"
  330 PRINT "RIGHT MOTOR FORWARD"
  340 PRINT "RIGHT MOTOR REVERSE"
  350 PRINT "LEFT MOTOR FORWARD"
  360 PRINT "LEFT MOTOR REVERSE"
  365 PRINT "LIGHTS"
  370 PRINT "PEN DOWN"
  380 PRINT "LOW TONE"
  390 PRINT "HIGH TONE"
  392 GOSUB 1000
  395 A=1
  400 POKE W,A
  410 GOSUB 1000
  420 A=2*A
  430 IF A=128 THEN A=192
  440 IF A<255 THEN 400
  450 POKE W,0
  500 PRINT
  510 PRINT "      INPUT TEST"
  520 PRINT "OPERATE MICRO SWITCHES BY TOUCHING THE DOME"
  525 FOR I=1 TO 8
  530 A= PEEK (R)
  540 IF A=247 THEN PRINT "BACK TOUCH"
  550 IF A=251 THEN PRINT "FRONT TOUCH"
  560 IF A=253 THEN PRINT "RIGHT TOUCH"
  570 IF A=254 THEN PRINT "LEFT TOUCH"
  580 IF A=255 THEN PRINT "NO TOUCH"
  585 PRINT "DATA IS ",A
  590 GOSUB 1000
  600 NEXT I
  610 END
 1000 REM  WAIT LOOP
 1010 C=0
 1020 C=C+1
 1030 IF C<300 THEN 1020
 1040 RETURN
```

```
PROGRAM EXAM;
(*USED TO LOOK AT AND MODIFY MEMORY LOCATIONS*)
(* PHILLIP MCKERROW  11.7.80 *)
VAR TERM,DATA,DAT1,ADDR:INTEGER;
    REQUEST1,REQUEST2,RUBISH:CHAR;
PROCEDURE POKE (DATA,ADDR:INTEGER);
EXTERNAL;
FUNCTION PEEK (ADDR:INTEGER):INTEGER;
EXTERNAL;
BEGIN
     TERM:=0;
     WHILE TERM=0 DO
     BEGIN
          WRITELN (OUTPUT,'ENTER REQUEST PEEK,POKE,QUIT');
          READ (INPUT,REQUEST1,REQUEST2);
          WHILE NOT EOLN (INPUT) DO READ (INPUT,RUBISH);
          READLN;
          IF REQUEST1='P' THEN
          BEGIN
               IF REQUEST2 = 'E' THEN
               BEGIN
                    WRITELN (OUTPUT,'ENTER ADDRESS - DECIMAL');
                    READLN (INPUT,ADDR);
                    DATA := PEEK(ADDR);
                    WRITELN (OUTPUT,'CONTENTS IS  ',DATA);
               END
               ELSE IF REQUEST2= 'O' THEN
               BEGIN
                    WRITELN(OUTPUT,'ENTER DATA 0-255, ADDRESS - DECIMAL
```

```
                    READLN (INPUT,DATA,ADDR);
                    POKE (DATA,ADDR);
                    DAT1:=PEEK(ADDR);
                    IF DAT1()DATA THEN WRITELN('ERROR ',DATA,'   ',DAT1);
                END;
            END
            ELSE IF REQUEST1 ='Q' THEN TERM := 1;
        END;
        WRITELN (OUTPUT,'BYE BYE');
END.

;ASSEMBLER ROUTINES TO CONTROL TURTLE
;THE UNIVERSITY OF WOLLONGONG
;PHILLIP MCKERROW   4.6.80
;PEEK AND POKE
;
;MACRO DEFINITIONS
;
;  MACRO POP ADDRESS
;POPS 16 BIT ARG FROM STACK TO ASSRESS
;
;
        .MACRO POP
        PLA             ;PULL LS BYTE
        STA %1
        PLA             ;PULL MS BYTE
        STA %1+1
        .ENDM
```

```
;  MACRO PUSH ADDRESS
;PUSHES 16 BIT ARG TO STACK
;
        .MACRO PUSH
        LDA %1+1
        PHA             ;PUT MS BYTE
        LDA %1
        PHA             ;PUT LS BYTE
        .ENDM
;
;
;       FUNCTIONS
;
;  FUNCTION PEEK(ADDRESS:INTEGER):INTEGER;
;RETURNS CONTENTS OF SPECIFIED ADDRESS
;8 BITS OF DATA RETURNED IN LS BYTE
;MS BYTE SET TO ZERO
;
        .FUNC PEEK,1
ADDR    .EQU 0      ;ADDRESS OF DATA
        POP RETURN          ;GET RETURN ADDRESS
        PLA                 ;STACK BIAS
        PLA
        PLA
        PLA
        POP ADDR            ;GET ADDRESS
        LDA #0
        PHA                 ;SET MS BYTE TO 0
        LDY #0
```

```
        LDA @ADDR,Y       ;GET DATA
        PHA               ;RETURN DATA
        PUSH RETURN
        RTS               ;GO BACK
RETURN  .WORD 0,0
;
;
;     PROCEDURE DEFFINITIONS
;
; PROCEDURE POKE(DATA,ADDR:INTEGER)
;PROCEDURE TO WRITE TO ADDRESS
;
        .PROC POKE,2
ADDR    .EQU 0
        POP RETURN    ;SAVE RETURN ADDRESS
        POP ADDR      ;MEMORY LOCATION
        LDX #0
        PLA           ;GET OUTPUT
        STA @ADDR,X   ;POKE
        PLA           ;CLEAN UP STACK
        PUSH RETURN
        RTS           ;GO BACK
RETURN  .WORD 0,0
        .END
```

INTERNAL CONTROLS

There are six locations within the PIA accessible to the MPU data bus: two Peripheral Registers, two Data Direction Registers, and two Control Registers. Selection of these locations is controlled by the RS0 and RS1 inputs together with bit 2 in the Control Register, as shown in Table 1.

TABLE 1 – INTERNAL ADDRESSING

RS1	RS0	Control Register Bit CRA-2	Control Register Bit CRB-2	Location Selected
0	0	1	X	Peripheral Register A
0	0	0	X	Data Direction Register A
0	1	X	X	Control Register A
1	0	X	1	Peripheral Register B
1	0	X	0	Data Direction Register B
1	1	X	X	Control Register B

X = Don't Care

INITIALIZATION

A low reset line has the effect of zeroing all PIA registers. This will set PA0-PA7, PB0-PB7, CA2 and CB2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

Details of possible configurations of the Data Direction and Control Register are as follows.

DATA DIRECTION REGISTERS (DDRA and DDRB)

The two Data Direction Registers allow the MPU to control the direction of data through each corresponding peripheral data line. A Data Direction Register bit set at "0" configures the corresponding peripheral data line as an input; a "1" results in an output.

CONTROL REGISTERS (CRA and CRB)

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA1, CA2, CB1 and CB2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occurring on control lines CA1, CA2, CB1 or CB2. The format of the control words is shown in Table 2.

TABLE 2 – CONTROL WORD FORMAT

	7	6	5	4	3	2	1	0
CRA	IRQA1	IRQA2	CA2 Control			DDRA Access	CA1 Control	
CRB	IRQB1	IRQB2	CB2 Control			DDRB Access	CB1 Control	

Data Direction Access Control Bit (CRA-2 and CRB-2) – Bit 2 in each Control register (CRA and CRB) allows selection of either a Peripheral Interface Register or the Data Direction Register when the proper register select signals are applied to RS0 and RS1.

Interrupt Flags (CRA-6, CRA-7, CRB-6, and CRB-7) – The four interrupt flag bits are set by active transitions of signals on the four Interrupt and Peripheral Control lines when those lines are programmed to be inputs. These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

TABLE 3 – CONTROL OF INTERRUPT INPUTS CA1 AND CB1

CRA-1 (CRB-1)	CRA-0 (CRB-0)	Interrupt Input CA1 (CB1)	Interrupt Flag CRA-7 (CRB-7)	MPU Interrupt Request $\overline{IRQA}$ ($\overline{IRQB}$)
0	0	↓ Active	Set high on ↓ of CA1 (CB1)	Disabled — $\overline{IRQ}$ remains high
0	1	↓ Active	Set high on ↓ of CA1 (CB1)	Goes low when the interrupt flag bit CRA 7 (CRB-7) goes high
1	0	↑ Active	Set high on ↑ of CA1 (CB1)	Disabled — $\overline{IRQ}$ remains high
1	1	↑ Active	Set high on ↑ of CA1 (CB1)	Goes low when the interrupt flag bit CRA-7 (CRB-7) goes high

Notes: 1. ↑ indicates positive transition (low to high)

2. ↓ indicates negative transition (high to low)

3. The interrupt flag bit CRA-7 is cleared by an MPU Read of the A Data Register, and CRB-7 is cleared by an MPU Read of the B Data Register.

4. If CRA-0 (CRB-0) is low when an interrupt occurs (interrupt disabled) and is later brought high, $\overline{IRQA}$ ($\overline{IRQB}$) occurs after CRA-0 (CRB-0) is written to a "one"

Control of CA1 and CB1 Interrupt Input Lines (CRA-0, CRB-0, CRA-1, and CRB-1) – The two lowest order bits of the control registers are used to control the interrupt input lines CA1 and CB1. Bits CRA-0 and CRB-0 are used to enable the MPU interrupt signals $\overline{IRQA}$ and $\overline{IRQB}$, respectively. Bits CRA-1 and CRB-1 determine the active transition of the interrupt input signals CA1 and CB1 (Table 3).

TABLE 4 – CONTROL OF CA2 AND CB2 AS INTERRUPT INPUTS
CRA5 (CRB5) is low

CRA-5 (CRB-5)	CRA-4 (CRB-4)	CRA-3 (CRB-3)	Interrupt Input CA2 (CB2)	Interrupt Flag CRA-6 (CRB-6)	MPU Interrupt Request $\overline{IRQA}$ ($\overline{IRQB}$)
0	0	0	↓ Active	Set high on ↓ of CA2 (CB2)	Disabled — $\overline{IRQ}$ remains high
0	0	1	↓ Active	Set high on ↓ of CA2 (CB2)	Goes low when the interrupt flag bit CRA-6 (CRB-6) goes high
0	1	0	↑ Active	Set high on ↑ of CA2 (CB2)	Disabled — $\overline{IRQ}$ remains high
0	1	1	↑ Active	Set high on ↑ of CA2 (CB2)	Goes low when the interrupt flag bit CRA-6 (CRB-6) goes high

Notes: 1. ↑ indicates positive transition (low to high)

2. ↓ indicates negative transition (high to low)
3. The Interrupt flag bit CRA-6 is cleared by an MPU Read of the A Data Register and CRB-6 is cleared by an MPU Read of the B Data Register.
4. If CRA-3 (CRB-3) is low when an interrupt occurs (Interrupt disabled) and is later brought high, $\overline{IRQA}$ ($\overline{IRQB}$) occurs after CRA-3 (CRB-3) is written to a "one".

TABLE 5 – CONTROL OF CB2 AS AN OUTPUT
CRB-5 is high

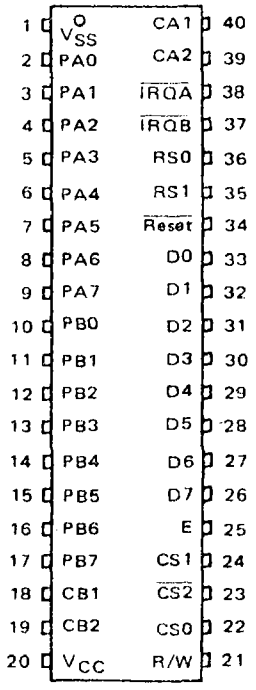
CRB-5	CRB-4	CRB-3	CB2 Cleared	CB2 Set
1	0	0	Low on the positive transition of the first E pulse following an MPU Write "B" Data Register operation.	High when the interrupt flag bit CRB-7 is set by an active transition of the CB1 signal.
1	0	1	Low on the positive transition of the first E pulse after an MPU Write "B" Data Register operation.	High on the positive edge of the first "E" pulse following an "E" pulse which occurred while the part was deselected.
1	1	0	Low when CRB-3 goes low as a result of an MPU Write in Control Register "B"	Always low as long as CRB-3 is low. Will go high on an MPU Write in Control Register "B" that changes CRB-3 to "one".
1	1	1	Always high as long as CRB-3 is high. Will be cleared when an MPU Write Control Register "B" results in clearing CRB-3 to "zero"	High when CRB-3 goes high as a result of an MPU Write into Control Register "B".

Control of CA2 and CB2 Peripheral Control Lines (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4, and CRB-5) – Bits 3, 4, and 5 of the two control registers are used to control the CA2 and CB2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA-5 (CRB-5) is low, CA2 (CB2) is an interrupt input line similar to CA1 (CB1) (Table 4). When CRA-5 (CRB-5) is high, CA2 (CB2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA2 and CB2 have slightly different characteristics (Tables 5 and 6).

TABLE 6 – CONTROL OF CA-2 AS AN OUTPUT
CRA-5 is high

CRA-5	CRA-4	CRA-3	CA2 Cleared	CA2 Set
1	0	0	Low on negative transition of E after an MPU Read "A" Data operation.	High when the interrupt flag bit CRA-7 is set by an active transition of the CA1 signal.
1	0	1	Low on negative transition of E after an MPU Read "A" Data operation.	High on the negative edge of the first "E" pulse which occurs during a deselect.
1	1	0	Low when CRA-3 goes low as a result of an MPU Write to Control Register "A".	Always low as long as CRA-3 is low. Will go high on an MPU Write to Control Register "A" that changes CRA-3 to "one".
1	1	1	Always high as long as CRA-3 is high. Will be cleared on an MPU Write to Control Register "A" that clears CRA-3 to a "zero".	High when CRA-3 goes high as a result of an MPU Write to Control Register "A".

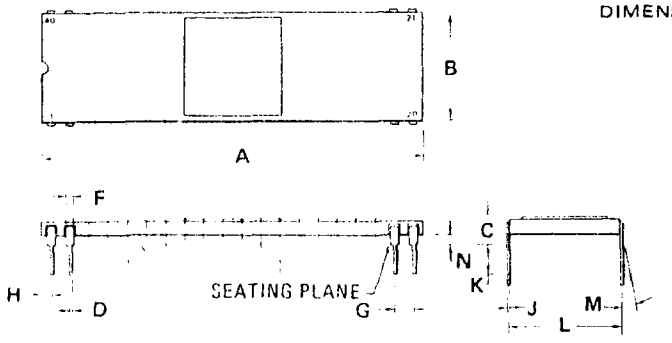
PIN ASSIGNMENT



PACKAGE DIMENSIONS

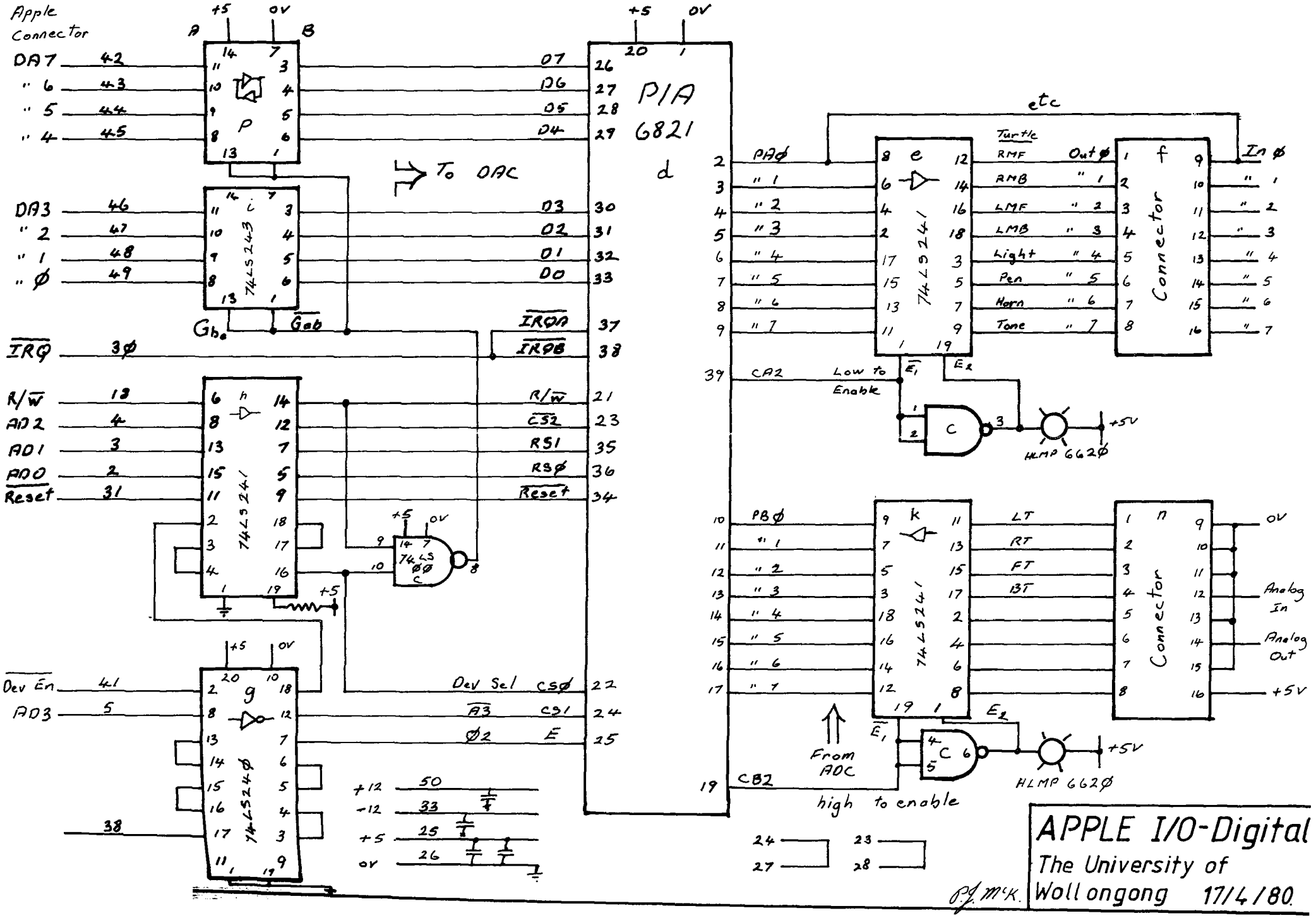
CASE 715-02
(CERAMIC)

SEE PAGE 165 FOR PLASTIC PACKAGE DIMENSIONS.



DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	50.29	51.31	1.980	2.020
B	14.86	15.62	0.585	0.615
C	2.54	4.19	0.100	0.165
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.33	0.008	0.013
K	2.54	4.19	0.100	0.165
L	14.60	15.37	0.575	0.605
M	-	10^0		10^0
N	0.51	1.52	0.020	0.060

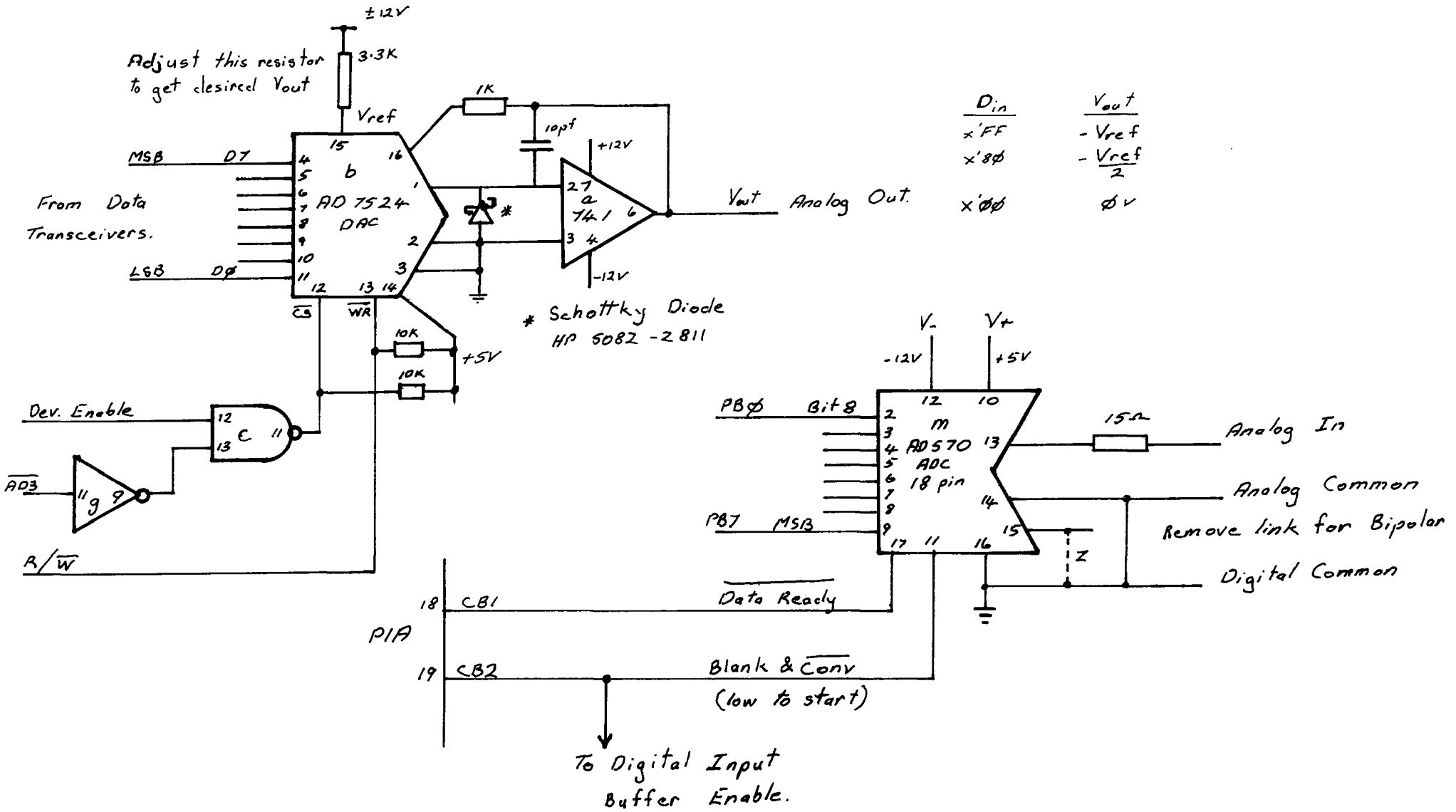
NOTE:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIA (AT SEATING PLANE), AT MAX. MAT'L CONDITION.



APPLE I/O-Digital

The University of Wollongong 17/4/80

P.J. McK.

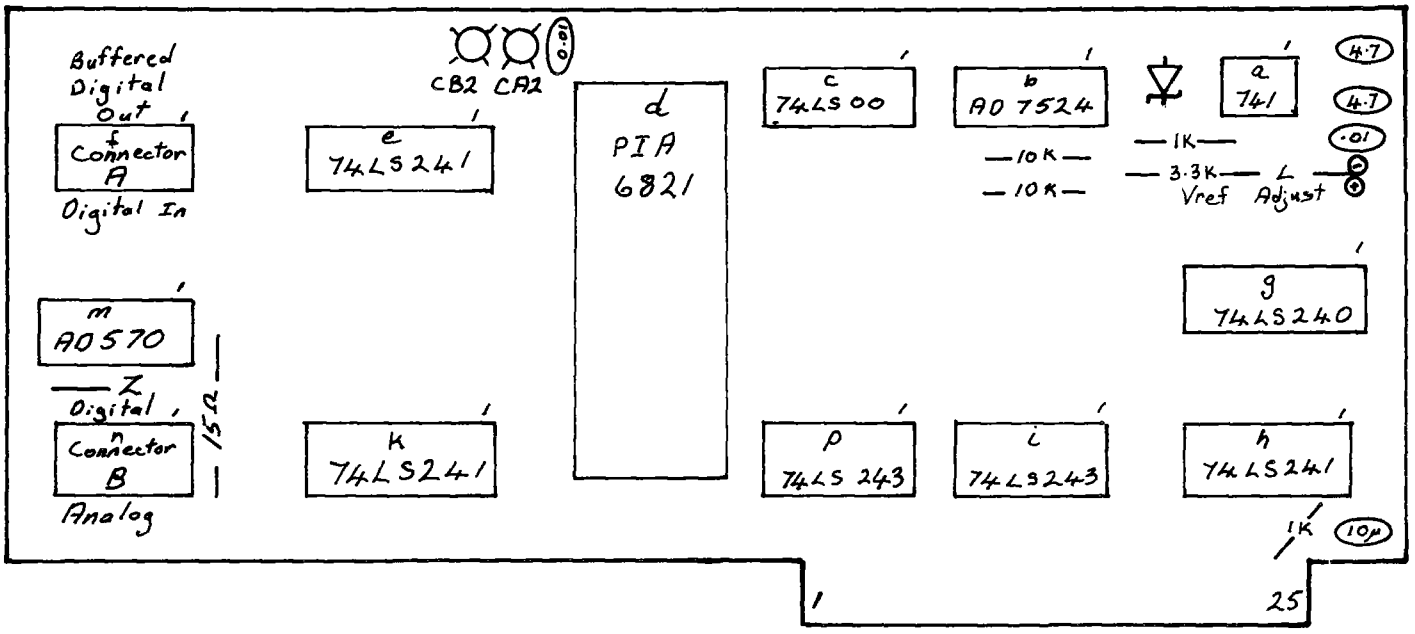


D_{in}	V_{out}
x'FF	$-V_{ref}$
x'80	$-\frac{V_{ref}}{2}$
x'00	0 V

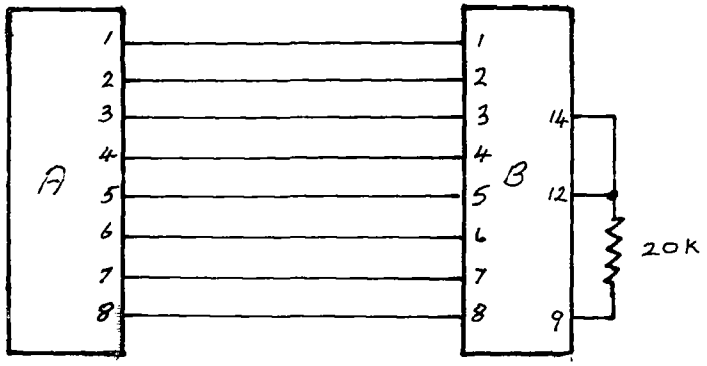
APPLE I/O-Analo

The University 18/4/80
of Wollongong P.J. M°K.

Component Overlay



Test Cable



APPLE I/O-Overlay
The University 18/4/8(
of Wollongong B.J. McK.